

Ahmat Barkai Moussa

Task 6 : Oluşturulan tablo için facade yazılması. Yeni tablo için yazılan facade ile birlikte populator yazılması ve converter ile Modelin Dataya convert edilmesi işlemini yapın. Homepage'de bir form oluşturun ve HomepageController'da bu facade'a istek atarak veriyi getirin, veriyi güncellemek için bir servis yazın ve formdan gelen verileri database'e kaydedin.

```
<bean class="com.avansas.facades.data.TrainingProductData">
  <property name="productName" type="java.lang.String"/>
  <property name="companyName" type="java.lang.String"/>
</bean>
```

Avansasfacade-beans.xml'de ekliyp ant build yapıyoruz .

```
Generated property for TrainingProductData.productName property defined at extension
avansasfacades.

private String productName;

Generated property for TrainingProductData.companyName property defined at extension
avansasfacades.

private String companyName;

public TrainingProductData()
{
    // default constructor
}

public void setProductName(final String productName)
{
    this.productName = productName;
}

public String getProductName()
{
    return productName;
}

public void setCompanyName(final String companyName) { this.companyName = companyName; }

public String getCompanyName() { return companyName; }
```

Data sınıfımız oluştu.

```

ler.java x landingLayout2Page.jsp x TrainingProductDao.java x TrainingProductDaoImp.java x avansasfacades-beans.xml x TrainingProduct
import de.hybris.platform.servicelayer.model.ModelService;
import de.hybris.platform.servicelayer.search.FlexibleSearchService;
import de.hybris.platform.servicelayer.search.SearchResult;

import javax.annotation.Resource;
import java.util.List;

// usage
public class TrainingProductDaoImp implements TrainingProductDao {

    // 1 usage
    private static final String FIND_ALL_EMPLOYEES = "SELECT {" + TrainingProductModel.PK + "} FROM {" + TrainingProductModel._TYPECODE + "}";

    // 2 usages
    @Resource(name="flexibleSearchService")
    private FlexibleSearchService flexibleSearchService;

    // no usages
    private ModelService modelService;

    // no usages
    @Override
    public List<TrainingProductModel> findAllProducts() {
        String query = "SELECT {pk} FROM {TrainingProduct}";
        final SearchResult<TrainingProductModel> result = flexibleSearchService.search(query);

        return result.getResult();
    }

    @Override
    public List<TrainingProductModel> getAllProducts()
    {
        final SearchResult<TrainingProductModel> result = flexibleSearchService.search(FIND_ALL_EMPLOYEES);
        return result.getResult();
    }
}

```

Sonradan doa interface ve class oluşturuyoruz. Query ve gereken metotlarımızı yazıyoruz.

```

<!-- Task 6 service and Dao bean integration -->

<!-- Service -->
<alias name="trainingProductService" alias="trainingProductService"/>
<bean id="trainingProductService" class="com.avansas.core.services.impl.TrainingProductServiceImp">
    <property name="trainingProductDao" ref="trainingProductDao"/>
</bean>

<!-- Dao -->
<alias name="trainingProductDaoImp" alias="trainingProductDao" />
<bean id="trainingProductDaoImp" class="com.avansas.core.daos.impl.TrainingProductDaoImp"/>

```

Dao ve service injection için gereken beans'leri avansascore-spring.xml'de yazıyoruz.

```

<!--Task 6 converter and populator configuration-->
<!--Converter-->
<alias name="trainingProductConverter" alias="trainingProductConverter"/>
<bean id="trainingProductConverter" parent="abstractPopulatingConverter">
  <property name="targetClass" value="com.avansas.facades.data.TrainingProductData"/>
  <property name="populators">
    <list>
      <ref bean="trainingProductPopulator"/>
    </list>
  </property>
</bean>
<alias name="trainingProductPopulator" alias="trainingProductPopulator"/>
<bean id="trainingProductPopulator" class="com.avansas.facades.populators.TrainingProductPopulator"/>

<!-- Reverse populating -->
<bean id="trainingProductReversePopulator" class="com.avansas.facades.populators.TrainingProductReversePopulator"/>
<bean id="trainingProductReverseConverter" parent="abstractPopulatingConverter">
  <property name="targetClass" value="com.avansas.core.model.TrainingProductModel"/>
  <property name="populators">
    <list>
      <ref bean="trainingProductReversePopulator"/>
    </list>
  </property>
</bean>

<!-- Facade -->
<alias name="trainingProductFacadeImp" alias="trainingProductFacadeImp"/>
<bean id="trainingProductFacadeImp" class="com.avansas.facades.product.impl.TrainingProductFacadeImp">
  <property name="trainingProductConverter" ref="trainingProductConverter"/>
  <property name="trainingProductReverseConverter" ref="trainingProductReverseConverter"/>
  <property name="trainingProductService" ref="trainingProductService"/>
</bean>

```

Facade , populator ve reverse populator için beans'leri yazıyoruz.

```

import javax.annotation.Resource;
import java.util.List;

3 usages
public class TrainingProductServiceImp implements TrainingProductService {

    2 usages
    @Resource(name="trainingProductDao")
    private TrainingProductDao trainingProductDao;

    3 usages
    @Resource(name="modelService")
    private ModelService modelService;

    @Override
    public List<TrainingProductModel> getAllProducts() {
        return trainingProductDao.getAllProducts();
    }

    1 usage
    @Override
    public TrainingProductModel updateTrainingProduct(TrainingProductModel trainingProductModel) {
        TrainingProductModel trainingProductModelNew = modelService.create(trainingProductModel.getClass());
        trainingProductModelNew.setProductName(trainingProductModel.getProductName());
        trainingProductModelNew.setCompanyName(trainingProductModel.getCompanyName());
        modelService.save(trainingProductModelNew);

        return trainingProductModelNew;
    }

    1 usage
    public void setTrainingProductDao(TrainingProductDao trainingProductDao) {
        this.trainingProductDao = trainingProductDao;
    }

    public void setModelService(ModelService modelService) {
        this.modelService = modelService;
    }
}

```

Sonra service interface ve servisi Imp yazılıyor .

```
import com.avansas.facades.product.TrainingProductFacade;
import de.hybris.platform.serviceLayer.dto.converter.Converter;

import javax.annotation.Resource;
import java.util.List;

1 usage
public class TrainingProductFacadeImp implements TrainingProductFacade {

    4 usages
    @Resource(name="trainingProductService")
    private TrainingProductService trainingProductService;

    3 usages
    @Resource(name="trainingProductConverter")
    private Converter<TrainingProductModel, TrainingProductData> trainingProductConverter;

    3 usages
    @Resource(name="trainingProductReverseConverter")
    private Converter<TrainingProductData, TrainingProductModel> trainingProductReverseConverter;

    @Override
    public List<TrainingProductData> getAllProducts() {
        return trainingProductConverter.convertAll(trainingProductService.getAllProducts());
    }

    1 usage
    @Override
    public TrainingProductData updateTrainingProduct(TrainingProductData trainingProductData) {
        TrainingProductModel trainingProductModel = trainingProductReverseConverter.convert(trainingProductData);
        trainingProductService.updateTrainingProduct(trainingProductModel);

        return getTrainingProductConverter().convert(trainingProductModel);
    }

    1 usage
    public Converter<TrainingProductModel, TrainingProductData> getTrainingProductConverter() {
        return trainingProductConverter;
    }
}
```

```
import com.avansas.facades.product.TrainingProductFacade;
import de.hybris.platform.serviceLayer.dto.converter.Converter;

import javax.annotation.Resource;
import java.util.List;

1 usage
public class TrainingProductFacadeImp implements TrainingProductFacade {

    4 usages
    @Resource(name="trainingProductService")
    private TrainingProductService trainingProductService;

    3 usages
    @Resource(name="trainingProductConverter")
    private Converter<TrainingProductModel, TrainingProductData> trainingProductConverter;

    3 usages
    @Resource(name="trainingProductReverseConverter")
    private Converter<TrainingProductData, TrainingProductModel> trainingProductReverseConverter;

    @Override
    public List<TrainingProductData> getAllProducts() {
        return trainingProductConverter.convertAll(trainingProductService.getAllProducts());
    }

    1 usage
    @Override
    public TrainingProductData updateTrainingProduct(TrainingProductData trainingProductData) {
        TrainingProductModel trainingProductModel = trainingProductReverseConverter.convert(trainingProductData);
        trainingProductService.updateTrainingProduct(trainingProductModel);

        return getTrainingProductConverter().convert(trainingProductModel);
    }

    1 usage
    public Converter<TrainingProductModel, TrainingProductData> getTrainingProductConverter() {
        return trainingProductConverter;
    }
}
```

Facade’da service’teki metotları çağırıyoruz , populator yada reverse populator işlemleri yapıyoruz.

```

TrainingProductFacadeImp.java x HomePageController.java x TrainingProductDaoImp.java x avansascore-spring.xml x TrainingProductData.java
@
@Resource(name = "cmsSiteService")
private CMSSiteService cmsSiteService;

1 usage
@Resource(name = "avsCommonB2BUnitService")
private AvsCommonB2BUnitService avsCommonB2BUnitService;

3 usages
@Resource(name = "trainingProductFacadeImp")
private TrainingProductFacade trainingProductFacade;

no usages
public static final String LOGOUT = "logout";

no usages new *
@RequestMapping(method = RequestMethod.POST)
public String updateTrainingProduct(final Model model, @RequestBody TrainingProductData trainingProductData){
    TrainingProductData trainingProductData1 = trainingProductFacade.updateTrainingProduct(trainingProductData);
    model.addAttribute("${trainingProductData1}",trainingProductData1);
    return getViewForPage(model);}

+ yigitdandin +9 *
@RequestMapping(method = RequestMethod.GET)
public String home(@RequestParam(value = "logout", defaultValue = "false") final boolean logout, @RequestParam(value = "message"
final RedirectAttributes redirectModel, final HttpServletResponse response, final HttpServletRequest request,
{
    // Training form set

    List<TrainingProductData> trainingProductData = trainingProductFacade.getAllProducts();

    TrainingProductForm trainingProductForm = new TrainingProductForm();

    trainingProductForm.setProductName(trainingProductData.get(1).getProductName());
    trainingProductForm.setCompanyName(trainingProductData.get(1).getCompanyName());
    model.addAttribute("${trainingProductData}",trainingProductFacade.getAllProducts().get(1));
}

```

HomePageController’de verileri alabilmemiz için Get metodu ve post metodu yazıp pageModel döndürüyoruz.

```

TrainingProductFacadeImp.java x HomePageController.java x landingLayout2Page.jsp x TrainingProductDaoImp.java x avansascore-spring.xml
<form:form id="trainingProductForm" action="/" method="post" var="trainingProductData" commandName="trainingProductData">
<div class="row">
<div class="col-sm-7 col-md-5">
<!-- Product Name Information Start -->
<label class="form-label">
<span class="form-title">
${productNameVar}
</span>
<input type="text" class="secondary-input justletter" id=""
name="productName" maxlength="1024"
data-msg-required="<spring:theme code='{This field is mandatory}' text='Cannot be empty'/">
placeholder="<spring:theme code='{product Name}' text='product Name'/" value="computer"
autocomplete="off"
required ${isB2C ? '' : 'readOnly'}>
<form:errors path="productName" cssClass="error" element="label" />
</label>
<!-- Product Name Information Finish -->
<!-- Company Name Information Start -->
<label class="form-label">
<span class="form-title">
${companyNameVar}
</span>
<input type="text" class="secondary-input justletter" id=""
name="companyName" maxlength="1024"
data-msg-required="<spring:theme code='{This field is mandatory}' text='Cannot be empty'/">
placeholder="<spring:theme code='{company Name}' text='company Name'/" value="casper"
autocomplete="off"
required ${isB2C ? '' : 'readOnly'}>
<form:errors path="companyName" cssClass="error" element="label" />
</label>
<!-- Company Name Information Finish -->
</div>
</div>
</form>

```

Ve Front end ‘de formumuzu ekleyip controller’de elde ettiğimiz (productName ve companyName) verileri form alanlarına ekliyoruz .

See the path again please

https://webztc.local:8002

OKULA DAIR HER ŞEY AVANSAS'TA! UYGUN KALİTE FİYAT

Avansas

Ürün, kategori veya ürün kodu bilgisi giriniz

ABA

0 ÜRÜN 0.00 TL

MAĞAZA VE BAŞKI (İLERİ) OFİS KURATILDI KAGİT ÜRÜNLERİ KARTUŞ - TONER TEMİZLİK VE SAĞLIK YİĞİT VE MUTFAK TEKNOLOJİ MOBİL VE YEREL

Training Product Form

productName

computer

companyName

casper

SavakKuyul

Güven Damgası Güvenli alışveriş

Hiçyenik Teslimat Alışveriş güvenliğiniz

Hızlı Teslimat 1 g içinde kargoya

Teslimat Beldava 100 TL ve üzeri alışverişiniz

Kolay İade 14 gün içerisinde

Siparişim Nerede? Siparişim güvenli

YENİ ÜRÜNLERİMİZİ DENEDİNİZ Mİ?

finish

2-1

Stalder

Ön yüzü formumuz verileriyle başarıyla geldi ve ayna anda güncelleyebilmektedir.

Commerce

Administration

types

SEARCH

3 Items

TrainingProduct.companyName	PK	TrainingProduct.productName
apple	8796093444100	mouse
casper	8796093345796	computer
apple	8796093313028	computer

SAVED QUERIES

No queries